

# A Formal Model for Context-Aware Semantic Augmented Reality Systems

Tamás Matuszka<sup>1,2(✉)</sup>, Attila Kiss<sup>1,3</sup>, and Woontack Woo<sup>2</sup>

<sup>1</sup> Eötvös Loránd University, Budapest, Hungary  
{tomintt,kiss}@inf.elte.hu

<sup>2</sup> KAIST, Daejeon, South Korea  
{tamas.matuszka,woo}@kaist.ac.kr

<sup>3</sup> J. Selye University, Komárno, Slovakia  
kissa@ujss.sk

**Abstract.** The Augmented Reality applications have received great attention in the recent years. However, there is still a lack of formal description of such systems currently. In this paper, we propose a new formal model for context-aware semantic Augmented Reality systems. The model can be divided into two parts: a set-theory function based method allows the formalization of an Augmented Reality system while an integrated time-space-motion logic provides the description of the behavior of the system. The suggested model enables the characterization of an Augmented Reality system with mathematical precision. In addition, logical inferences can be performed by means of the logic part of the formal description. The practical applicability of the proposed model is shown through use cases.

**Keywords:** Augmented reality · Semantic web · Context-aware computing · Formal model

## 1 Introduction

Augmented Reality (AR) [1] systems have become the accepted part of our everyday life due to the proliferation of smartphones. The physical environment of a user can be extended by virtual elements using such a system. An Augmented Reality system is regarded as a representative of context-aware computing [2], where the continuously changing context and the environment are taken into account by the system. As a result, the user is able to obtain and visualize context-dependent information.

The behavior of the state-of-the-art Augmented Reality systems can only be described as architectural diagrams nowadays. This fact raises the following question: how can we formally model a context-aware semantic augmented reality system? A formal description is needed that is able to model the context-aware AR systems with mathematical precision. In addition, the features of a context-aware Augmented Reality system can be described using the formalism.

In this paper, a new formal model for context-aware semantic Augmented Reality systems is presented. The model consists of two parts. The components of the AR system are formalized by the first part of the model using set theory functions.

The second part can be used for describing the behavior of the system with integrated space-time-motion logic. Therefore, the system enables to execute logical inferences. Our model contributes to the development of theoretical foundations of Augmented Reality systems. For validation purposes, two context-aware mobile Augmented Reality browsers have been implemented based on our abstract formal model.

The structure of the paper is as follows. After the introductory Sect. 1, related work is presented in Sect. 2. The proposed formal model is described in Sect. 3. Then, use cases where the formal model was applied are introduced in Sect. 4. Finally, the conclusion and future work are shown in Sect. 4.

## 2 Related Work

In the last decades, several definitions of Augmented Reality have been developed. One of the most popular definitions is given by Azuma [1], who described Augmented Reality in an informal way using three characteristics. Milgram and Kishino [3] defined a taxonomy of mixed reality and introduced the concept of virtuality continuum which positioning Augmented Reality between the real environment and the virtual environment. Despite the popularity of these definitions, those are still informal explanations and a formal description is needed in order to ensure the mathematical precision.

Reicher introduced a framework for AR systems in his doctoral thesis [4]. A reference architecture and design patterns for Augmented Reality have also been described in that work. The proposed method provides a detailed description of a wearable Augmented Reality system architecture and can be used as a guideline for AR system design. However, the solution does not contain any formal model or description.

Galton developed a temporal logic in [5] which is able to describe time-dependent and spatial phenomena. The work is based on the spatial logic introduced by Randell, et al. [6]. Owing to the capability of the description of continuous movement, this framework can be used for modeling the behavior of an Augmented Reality system. In addition, reasoning about the motion of a user of the AR system is also become available by means of the logic. The details can be seen in Sect. 3.

## 3 CAESAR Model

The formal description of the proposed system is introduced in this section. The model has been called as CAESAR (Context-Awareness Enriched Semantic Augmented Reality) and consists of a set theory functions-based part that formalizes the components of an AR system while an integrated time-space-motion logic is used to describe the behavior of the system.

### 3.1 Formal Description with Set Theory Functions

The formal description of each component of CAESAR model is described in the following section using set theory functions. Furthermore, the formalization of the features provided by CAESAR model can also be seen in this subsection. The model includes the following four components:

- data: provides the user-generated and the integrated POI data from different data sources that are displayed by the AR browser,
- browser: is responsible for the visualization of data (provided by data component),
- semantic: is responsible for the connection to semantic web,
- context: adaptive segment that recommends content based on the context.

Formally, a quartet  $CAESAR := \langle Data, ARB, SWB, C \rangle$  is called as a context-aware semantic Augmented Reality system, where  $Data$ ,  $ARB$ ,  $SWB$ , and  $C$  are the components of data, browser, semantic and context, respectively.

**Data Component.** The first building block of the model is the data component which is responsible for the displayable data provision. The data can be derived from different sources that use different storage schemas. Therefore, heterogeneous data integration is needed which includes the global schema matching and entity resolution.

**Definition 1.** *The triple  $\mathcal{G}, \mathcal{S}, \mathcal{M}$  is called as data integration system, where  $\mathcal{G}$  is the global schema,  $\mathcal{S}$  is the set of source elements, and  $\mathcal{M}$  is the mapping among the global schema and the schemas of the heterogeneous data.*

A requested query  $q$  will be queried against the integrated data source using the global schema  $\mathcal{G}$ . During this method,  $\mathcal{M}$  is responsible for mapping  $q$  to  $\mathcal{S}$  [7].

Schema matching can be used for determining global schema  $\mathcal{G}$  based on  $\mathcal{S}$ . The core element of schema matching is operator  $match_{SM}$ . The definition of  $\mathcal{M}_{SM}$  which is the mapping between two schemas is needed before introducing  $match_{SM}$ .

**Definition 2.** *The mapping  $\mathcal{M}_{SM}$  is the set of mapping elements between schema  $S_1$  and  $S_2$ . The mapping elements represent that certain elements of  $S_1$  have been mapped into certain elements of  $S_2$ .*

**Definition 3.** *The operator  $match_{SM}$  is a function  $f : \mathcal{S} \times \mathcal{S} \rightarrow \mathcal{M}_{SM}$  which creates the mapping  $\mathcal{M}_{SM}$  from the given two schemas. The resulted overcome is called as match result [8].*

The identification of duplicated elements is crucial during the data integration process and entity resolution can be used for this purpose. Function  $match_{ER}$  (which is different from the above-mentioned  $match_{SM}$ ) can be used in order to identify the same elements (in our case, the POIs which represent the same real-world entity).

**Definition 4.** *Let  $E$  be the set of entities. Then,  $match_{ER}$  is a Boolean function  $f : E \times E \rightarrow \{true, false\}$  which determines whether two entities are matching (i.e. represent the same real-world entity) or not (denoted by  $e_1 \approx e_2$ , if  $match(e_1, e_2) = true$ , where  $e_1, e_2 \in E$ ).*

A partial order on the entities can be defined using the entity-related useful information. If entity  $e_2$  holds more information than  $e_1$ , then  $e_2$  dominates  $e_1$  (denoted by  $e_1 \preceq e_2$ ).

**Definition 5.** The function  $\mu : E \times E \rightarrow E$  (called as merge) merges two matching  $e_1 \approx e_2$  entities into one entity. During the merging method, the function keeps only the dominant entity  $e_2$ , and extends it with the missing attributes deriving from  $e_1$ .

**Definition 6.** An instance  $I = \{e_1, \dots, e_n\}$  is a finite set of entities from  $E$ .

The merge closure finds all matching entities within instance  $I$  and merges them using the match and merge functions.

**Definition 7.** Let  $I$  be an instance, then merge closure of  $I$  (denoted by  $\bar{I}$ ) the smallest set of  $S$  such that  $I \subseteq S$ . In addition,  $\forall e_1, e_2 \in S$ , if  $e_1 \approx e_2$ , then  $\text{merge}(e_1 \approx e_2) \in S$ .

The domination of entities can naturally be extended to the instances as well.

**Definition 8.** Let  $I_1, I_2$  be two instances. Then,  $I_1$  is dominated by  $I_2$  (denoted by  $I_1 \preceq I_2$ ), if  $\forall e_1 \in I_1, \exists e_2 \in I_2$ , such that  $e_1 \preceq e_2$ .

The definition of entity resolution can be defined using the above auxiliary formulas.

**Definition 9.** Let  $I$  be an instance,  $\bar{I}$  be the merge closure of  $I$ . An entity resolution of  $I$  is the  $I'$  set of entities such that  $I' \subseteq \bar{I}$  and  $\bar{I} \not\preceq I'$ . In addition, there is not any proper subset of  $I'$  which satisfies the first two conditions [9].

In conclusion, component *Data* is a  $\mathcal{G}, \mathcal{S}, \mathcal{M}$  data integration system which creates global schema  $\mathcal{G}$  from source schemas  $\mathcal{S}$  using operator  $\text{match}_{SM}$ . Furthermore, the execution of entity resolution during the data integration phase is also the responsibility of component *Data*.

**Browser Component.** The second component of CAESAR model is the browser component, which is responsible for the displaying of the data provided by the before mentioned data component. The visualization uses Augmented Reality, which can be formally defined in the following way.

**Definition 10.** A quintet  $\mathcal{M}, \mathcal{VE}, \mathcal{T}, \varphi, \xi$  is called as Augmented Reality system, where  $\mathcal{M}$  is the set of the markers,  $\mathcal{VE}$  is the set of the virtual elements,  $\mathcal{T}$  is the set of transformations,  $\varphi$  is the mapping function, and  $\xi$  is the transformation function.

Let  $IB, PB$  (image-based markers and position-based markers) be two disjoint sets. Then,  $\mathcal{M}$  can be written as follows:

$$\mathcal{M} = IB \cup PB. \quad (1)$$

Let  $I, V, S$  and  $K$  (images, videos, sounds, knowledge base, respectively) be pairwise disjoint sets. Then the set of virtual elements  $\mathcal{VE}$  can be written in the next form:

$$\mathcal{VE} = I \cup V \cup S \cup K. \quad (2)$$

The set  $\mathcal{T}$  contains geometric transformations, namely translation ( $\tau$ ), rotation ( $\rho$ ) and scale ( $\sigma$ ). In addition, let  $L$  be the set of 3D vectors. Every virtual element  $v \in \mathcal{VE}$  has an  $l \in L$  vector. The vector  $l$  stores the position of virtual element  $v$ .

Function  $\varphi : \mathcal{M} \rightarrow \mathcal{VE} \times L$  maps a virtual element and its relative initial position to a marker. The range of function  $\varphi$  contains the empty set (i.e. there is not any virtual element assigned to a given marker).

The last part of the quintet is the transformation function  $\xi$ . Function  $\xi : \mathcal{M} \times \mathcal{VE} \times L \times \mathcal{T} \rightarrow \mathcal{VE} \times L$  transforms a virtual element corresponding to the given marker with a given transformation in real-time.

The current Augmented Reality systems can be modeled by the above-mentioned definitions. The browser component of CAESAR model is an AR system  $\langle \mathcal{M}|_{\text{PB}}, \mathcal{V}|_{\text{KB}}, \mathcal{T}, \varphi, \xi \rangle$  where set  $\mathcal{M}$  is restricted to the position-based markers and set  $\mathcal{VE}$  is restricted to knowledge base.

**Semantic Component.** The third component of the model is based on semantic web technologies and is responsible for the interconnection of data component provided information with publicly available semantic data sources. Since semantic data sources are interlinked using IRIs (*Internationalized Resource Identifier*), therefore, the semantic data source can be explored starting from a concept derived from the data component.

Let  $B, I$ , and  $L$  be the pairwise disjoint sets of blank nodes, literals and IRIs, respectively. In addition, let  $BIL, BI$ , and  $IL$  be the abbreviations of  $B \cup I \cup L, B \cup I$ , and  $I \cup L$ , respectively. The abbreviation  $BIL$  can be referred as RDF term.

**Definition 11.** A triplet  $(s, p, o) \in BI \times I \times BIL$  is called as RDF triple, where predicate  $p$  connects subject  $s$  with object  $o$ . Let the finite set of RDF triple be denoted by  $RDF_3$ .

**Definition 12.** Let the infinitive set of RDF triples be denoted by  $RDF_{DB}$  and be called as RDF database (or RDF document).

The data storage model of semantic web can be described by Definition 11 and 12. However, a query language is needed in order to access and manipulate the data stored in such a way. One of the semantic technologies, SPARQL query language was designed for this purpose. Since the RDF triples can be considered as directed edges, the RDF database can be seen as a directed graph. Due to this solution, the SPARQL language reduces the RDF database searching to graph pattern matching [10].

Ontology is one of the key components of semantic web, which describe the relations, rules and restrictions among the concepts.

**Definition 13.** An ontology is a structure  $\mathcal{O} := (C, \leq_C, P, \sigma)$ , where  $C$  and  $P$  are the disjoint sets of classes and properties, respectively. A partial order  $\leq_C$  on  $C$  is called as class hierarchy while the function  $\sigma : P \rightarrow C \times C$  describes the signature of properties. Let  $c_1, c_2 \in C$  be two classes. If  $c_1 \leq_C c_2$ , then  $c_1$  is the subclass of  $c_2$  while  $c_2$  is the superclass of  $c_1$  [11].

The following definitions will be used in the description of the semantic component of CAESAR model.

**Definition 14.** Let link  $: RDF_3 \times RDF_3 \rightarrow \{true, false\}$  be a Boolean function, which decides whether two RDF triples are directly accessible from each other or not. Let  $\rightsquigarrow$  denote this function.

The behavior of function *link* can formally be described in the following way:

$$(s_1, p_1, o_1) \rightsquigarrow (s_2, p_2, o_2) = \begin{cases} \text{true, if } s_1 \neq s_2 \wedge (o_1 = s_2 \vee o_2 = s_1 \vee (o_1 = o_2 \wedge o_1 \notin L)) \\ \text{false, otherwise} \end{cases}, \quad (3)$$

where  $(s_1, p_1, o_1), (s_2, p_2, o_2) \in \text{RDF}_3$ .

The directly accessible (one step relation) RDF triples can be obtained using function *link*. The indirectly accessible RDF triples can be determined using a method which is similar to the derivation method of formal languages theory.

**Definition 15.** Let  $\text{link}_{\text{indirect}^*} : \text{RDF}_3 \times \dots \times \text{RDF}_3 \rightarrow \{\text{true}, \text{false}\}$  be a Boolean function which determines whether two RDF triples are indirectly accessible from each other or not. Let  $\rightsquigarrow_*$  denote this function.

A  $b \in \text{RDF}_3$  is indirectly accessible or derivable from  $a \in \text{RDF}_3$  triple, if the following condition holds:

$$\exists n \in \mathbb{N}, r_1, \dots, r_n \in \text{RDF}_3 : a = r_1 \wedge b = r_n \wedge (\forall i \in [1 \dots n - 1] : r_i \rightsquigarrow r_{i+1}). \quad (4)$$

The component *SWB* is a semantic database which satisfies the following criteria:

$$\exists d \in \text{Data}, \forall r \in \text{RDF}_{\text{DB}} : d \rightsquigarrow_* r. \quad (5)$$

In details, at least one element from the information provided by data component can be used to explore the full semantically represented dataset.

**Context Component.** The last component of the model, which is responsible for the contextual data tailoring. Since the data provision has to take into account various parameters (location, time, etc.), this component can be seen as a context-aware recommender system that is based on [12]. The first step of the recommendation process is the specification of the set of initialization of recommendations which can be done by explicit (given by the users) or implicit (by means of inferences) way. After the initial data source has been created, the component tries to estimate the following recommendation function *R*:

$$R : \text{User} \times \text{POI} \times \text{Context} \rightarrow \text{Rating}, \quad (6)$$

where *User* is the set of users, *POI* the data coming from component *Data*, *Context* is the set of contextual information while *Rating* is the range of the ratings.

When function *R* has performed the estimation over  $\text{User} \times \text{POI} \times \text{Context}$  space, the context component returns the POI which has the highest *Rating* value. Intuitively, the component tries to estimate a value for unknown POIs. Function *R* can be defined in the following way.

**Definition 16.** Let  $R : \text{User} \times \text{POI} \times \text{Location} \times \text{Time} \times \text{Category} \rightarrow \text{Rating}$  be a recommender function which determines that the  $u \in \text{User}$  how would rate the  $p \in \text{POI}$  (deriving from data component) which is located in  $l \in \text{Location}$  and has a  $c \in \text{Category}$  in  $t \in \text{Time}$ , namely  $R(u, p, l, c, t) \in \text{Rating}$ .

**Formal Description of the System Functionality.** After the introduction of the components of the model, the abstract description of the features provided by CAESAR model can be read in this subsection. The definition of multivalued functions [13] is needed for the formal description of the features.

**Definition 17.** Let multivalued function  $f : A \rightarrow B^*$  a function which assigns one or more value from range to the elements of the domain, namely  $\forall x \exists n \in \mathbb{N}, y_1, \dots, y_n, \forall i \in [1, n] : (x, y_i) \in f$ .

The signatures of the features of CAESAR model can be defined by means of set theory functions and Definition 17. The core function of the model is function *browse* :  $Data \times String \times \mathbb{R} \rightarrow \mathcal{VE}^*$ , which collects the virtual elements that represents the relevant hits from the POI data derived from component *Data*, from search keyword and from the search radius (the current position of user is considered as the central point of search area). The resulted overcome (e.g. a POI) can be visualized by component *ARB*. In addition, detailed information can be displayed from a given POI using function *details* :  $\mathcal{VE} \rightarrow I \times String^n$ , where  $I$  is the set of images while the second component of the range is an  $n$ -dimensional word vector which represents the attributes of virtual element. A new virtual element can also be created by component *ARB* using function *new* :  $\mathbb{R} \times \mathbb{R} \times I \times String^n \rightarrow \mathcal{VE}$ . The inputs of this feature are the latitude and longitude coordinates, an image, and a word vector which contains the properties of a POI. Furthermore, navigation is also provided by component *ARB* by means of function *navigate* :  $\mathcal{VE} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow (\mathbb{R} \times \mathbb{R})^*$ . The function *navigate* gets a virtual element (POI) and its latitude/longitude coordinates, and the current latitude/longitude coordinates of a user and returns with the latitude/longitude coordinate pairs which represents the path to the given POI from the current position. Function *toRDF* :  $\mathcal{VE} \times \mathbb{R}^n \rightarrow RDF_3^*$  creates the RDF representation of a virtual element. In this way, a POI can be used for exploring a semantic dataset using Definition 15 and Eq. 5.

### 3.2 Integrated Time-Space-Motion Logic

The logical description of our CAESAR system is based on Galton's work [5] which introduces a special logic that enables to combine temporal and spatial logics. In this way, the changes in context can be described and logical inferences can be executed.

Galton's logic is based on the work of Randell et al. [6] who developed a spatial reasoning system (hereinafter referred to as RCC). RCC determines the set of relations between spatial regions and the concept of connection is considered as a primitive.

The concept of connection can be extended to spatial points beside the regions. In case of points, connection can take two values: two points either are the same points or different points. The relation between a point and a region can be the followings:

- $p$  is inside  $r$ ,
- $p$  bounds  $r$ ,
- $p$  is outside  $r$ .

*Inside* can be considered as a primitive, then, *Bounds* and *Outside* can be defined in the following way:

$$Bounds(p, r) \equiv \forall r' (Inside(p, r') \rightarrow PO(r, r')) \quad (7)$$

$$Outside(p, r) \equiv \neg Inside(p, r) \wedge \neg Bounds(p, r), \quad (8)$$

where  $PO$  means partially overlapping. The fulfillment of certain conditions can be indicated by three predicates, namely *Holds – on*, *Holds – in*, and *Holds-at*. The first two predicates deal with the state of intervals while the last can be used for relating states to instants. In addition, let  $inf(i)$  and  $sup(i)$  denote the boundaries of an interval  $i$  while let  $Div(t, i)$  denote that instant  $t$  falls within interval  $i$ .

Events can be defined using the changes of states. Similarly to the before mentioned predicate *Holds*, three *Occurs* predicates can be introduced, namely *Occurs – on*, *Occurs – in*, and *Occurs – at*. If an event  $e$  occurs during interval  $i$ , then predicate *Occurs – in*( $e, i$ ) can be used. If event  $e$  takes the whole interval  $i$ , *Occurs – on*( $e, i$ ) should be used. If event  $e$  is instantaneous and occurs at the instant  $t$ , then *Occurs – at*( $e, t$ ) can be written [5].

**Logical Formalization of the System.** The first step of the logical description is to define the following predicates and function symbol:

- *Region*( $x$ ) –  $x$  is a variable with type region,
- *VirtualRegion*( $x$ ) –  $x$  is a variable with type virtual region,
- *Point*( $x$ ) –  $x$  is a variable with type point,
- *VirtualPoint*( $x$ ) –  $x$  is a variable with type virtual point,
- *ARDevice*( $x$ ) –  $x$  is a variable with type AR device,
- *POI*( $x$ ) –  $x$  is a variable with type POI,
- *Belongs*( $dev, vr$ ) – indicates that virtual region  $vr$  belongs to AR device  $dev$ ,
- *Display*( $dev$ ) – indicates, whether AR displaying is possible on AR device  $dev$ ,
- *Type*( $x$ ) – function symbol, returns the category of variable  $x$  with type POI (e.g. university or restaurant).

The device which enables the AR displaying can be indicated by predicate *ARDevice*( $x$ ). Predicate *VirtualRegion*( $x$ ) represents a virtual region that belongs to variable  $y$  with type *Region* while *VirtualPoint*( $x$ ) represents a virtual element.

Hereinafter, the terms of virtual world and virtual region are used interchangeably. A variable with type virtual point can correspond to a POI variable; therefore, a virtual element can be assigned to a real world object. The following formalism can be used for creating a hierarchy among the types which is necessary for the further description:

$$\forall x (VirtualPoint(x) \rightarrow Point(x)) \quad (9)$$

$$\forall x (POI(x) \rightarrow Point(x) \vee Region(x)) \quad (10)$$

$$\forall x (ARDevice(x) \rightarrow Point(x) \vee Body(x)) \quad (11)$$

$$\forall x (VirtualRegion(x) \rightarrow Region(x)) \quad (12)$$



In possession of the above predicates and statements, the logical description of a CAESAR system can be specified in more detail. The first building component is the assignment of physical region with the corresponding virtual region. For this purpose, a formula is needed which decides whether a point belongs to a region, formally:

$$\begin{aligned} \text{Contains}(r, p) &\equiv \text{Region}(r) \wedge \text{Point}(p) \\ &\quad \wedge (\text{Bounds}(\text{pos}(p), r) \vee \text{Inside}(\text{pos}(p), r)) \end{aligned}$$

where  $\text{pos}(p)$  is the position of point  $p$ . The correspondence among the real and virtual points can be written using the above formula in the following way:

$$\begin{aligned} \text{Superimpose}(r, vr) & \\ &\equiv \forall \text{poi}(\text{POI}(\text{poi}) \wedge \text{Contains}(r, \text{poi})) \\ &\rightarrow \exists \text{vpoi}(\text{VirtualPoint}(\text{vpoi}) \wedge \text{Contains}(vr, \text{vpoi}) \wedge \text{pos}(\text{poi}) \\ &= \text{pos}(\text{vpoi})) \end{aligned}$$

Therefore, every physical POI belonging to region  $r$  can be assigned to a virtual object belonging to virtual region  $vr$  (which is corresponding to region  $r$ ). The next step is the correspondence between the real and virtual world such that the virtual world corresponds to the given AR device. In this way, the physical region can correspond to the virtual region which can be seen on the display of the AR device:

$$\begin{aligned} \text{ARMode}(dev, r, vr) & \\ &\equiv (\text{ARDevice}(dev) \wedge \text{Region}(r) \wedge \text{VirtualRegion}(vr) \\ &\quad \wedge \text{Belongs}(dev, vr) \wedge \text{pos}(vr) = \text{pos}(r) \wedge \text{Superimpose}(r, vr)) \\ &\rightarrow \text{Display}(dev). \end{aligned}$$

The formula checks whether virtual region  $vr$  belonging to AR device  $dev$  can correspond to the given region  $r$ . In addition, it examines the correspondence of physical POIs with virtual elements. If both conditions hold, then AR device  $dev$  is able to superimpose AR content into region  $r$ , i.e. the browser component *ARB* of CAESAR model can be used in the region  $r$ .

Certain questions related to the behavior of an AR system can be answered using logical inferences on data derived from the semantic *SWB* and the context *C* components of CAESAR model.

Such a behavior related question is that whether virtual content is available passing through a region. The answering logical formula is the following:

$$\begin{aligned} \text{Occurs-on}(\text{transitAR}(p, r, dev, vr), i) & \\ &\equiv \text{Holds} - \text{on}(\text{pos}(p) = \text{pos}(dev), i) \wedge \text{Holds} \\ &\quad - \text{at}(\text{Bounds}(\text{pos}(p), r), \text{inf}(i)) \wedge \text{Holds} \\ &\quad - \text{at}(\text{Bounds}(\text{pos}(p), r), \text{sup}(i)) \wedge \text{Holds} \\ &\quad - \text{on}(\text{Inside}(\text{pos}(p), r) \wedge \text{ARMode}(dev, r, vr), i), \end{aligned}$$

where  $transitAR(p, r, dev, vr)$  describes the event which represents the motion of a user during the AR system usage. The position of AR device  $dev$  and the moving point  $p$  which represents the user is the same during the whole interval of passing through. Point  $p$  bounds one of the boundaries of region  $r$  in the first instant of passage, it crosses the opposite boundary in the last instant while it is located inside of region  $r$  in the meantime. Predicate  $ARMode$  contained by the last  $Holds - on$  predicate determines whether the displayable virtual content is available.

The above question is related to the virtual elements within a region; however, it is independent of time. The need to respond to issues where the time is essential was also raised. The introduced logical description can be used to answer time-related questions as well. Let us consider the following two questions:

- Does the user miss a meeting at a given time and location?
- Is the user allowed to enter a place at a given time?

The first question can be answered using the following logical formula:

$$\begin{aligned} Occurs-at(contact(b_1, b_2, p_{loc}, t), t_{cur}) &\equiv t_{cur} \\ &\leq t \wedge Holds \\ &- at((EC(pos(b_1), pos(b_2)) \wedge pos(b_1) = pos(p_{loc})), t_{cur}) \\ &\wedge \exists t'(Holds - on(DC(pos(b_1), pos(b_2)), (t', t_{cur}))). \end{aligned}$$

The meeting of users  $b_1$  and  $b_2$  is represented by event  $contact(b_1, b_2, p_{loc}, t_{cur})$ . The meeting time and place have been agreed in location  $p_{loc}$  at time  $t$ . The truth value of the formula is true if the current time less than or equal to time  $t$ ;  $b_1$  and  $b_2$  are externally connected in location  $p_{loc}$  at time  $t$  (i.e. the meeting has happened), and there is a time  $t'$  when  $b_1$  and  $b_2$  are disconnected. The truth value of the following formula answers the second question:

$$\begin{aligned} Occurs-in(enter(b, r), i) \\ &\equiv POI(r) \wedge \exists t'(t' \geq \inf(i) \wedge Holds - at(EC(pos(b), r), t')) \\ &\wedge \exists t''(Div(t'', i) \wedge t'' > t' \wedge Holds - at(EC(pos(b), r), t'')) \\ &\wedge \exists t(Div(t, (t', t'')) \wedge Holds - on(PO(pos(b), r), (t', t''))). \end{aligned}$$

A user  $b$  is permitted to pass through a region  $r$  if the entering time falls within the allowed time interval  $i$  and the exit time falls also within this interval. Since the type of region  $r$  is POI (see Eq. 10), the related opening/allowed time interval can be obtained from data component *Data* or semantic component *SWB* of CAESAR model.

The last example is the formalization of navigation. The navigation regarded as a success if the user gets to point  $B$  from point  $A$  passing through the path made up of regions. The formalization requires defining the passage of point  $p$  through region  $r$ :

$$\begin{aligned}
\textit{Occurs} - \textit{on}(\textit{transit}(p, r), i) \\
&\equiv \textit{Holds} - \textit{at}(\textit{Bounds}(\textit{pos}(p), r), \textit{inf}(i)) \wedge \textit{Holds} \\
&\quad - \textit{at}(\textit{Bounds}(\textit{pos}(p), r), \textit{sup}(i)) \wedge \textit{Holds} \\
&\quad - \textit{on}(\textit{Inside}(\textit{pos}(p), r), i).
\end{aligned}$$

The logical statement of navigation can be given using the above formula:

$$\begin{aligned}
\textit{Occurs} - \textit{on}(\textit{navigate}(p_{\textit{user}}, r_1, \dots, r_n, a, b), i) &\equiv r_1 = a \wedge r_n = b \wedge \forall r_{\textit{wp}} \\
&\in [r_1, \dots, r_n] : (\textit{POI}(r_{\textit{wp}}) \wedge \textit{Occurs} - \textit{in}(\textit{transit}(p_{\textit{user}}, r_{\textit{wp}}), i) \\
&\quad \wedge \exists \textit{vpoi}(\textit{VirtualPoint}(\textit{vpoi}) \wedge \textit{Inside}(\textit{pos}(\textit{vpoi}), r_{\textit{wp}}))).
\end{aligned}$$

A point  $p_{\textit{user}}$  which represents the user passes through each region  $r_{\textit{wp}}$  (i.e. every waypoint of the navigation path). Since the type of the waypoints of navigation path is POI, virtual elements can be assigned to the waypoints and these points can be displayed by Augmented Reality, helping the user during the navigation. If time constraints are also available for each waypoint, then predicate  $\textit{Occurs} - \textit{in}(\textit{enter}(p_{\textit{user}}, r_{\textit{wp}}), i)$  can be used. In this way, the handling of time, space, and motion changes provided by the logic is fully utilized during the logical inference.

## 4 Use Cases

Two context-aware Android-based mobile AR browsers have been implemented which are based on our formal model. The architecture of the developed applications has been mapped to the proposed formal model, and the described modules have been also included. The first one is a tourist application, it allows the users to collect and display POIs in the surrounding environment [13]. The second is used in the field of cultural heritage; it provides movable story maps based on time and space [14]. Both applications use integrated data sources coming from component *Data* and this information was visualized by component *ARB*. In the case of the first application, publicly available semantic data sources can be explored starting from a given POI using component *SWB* while the second application provides POI-related semantic metadata browsing by means of component *SWB*. The component *C* was restricted to location, time and POI's category. These applications can be seen as a proof of the practical applicability of our formal model. The detailed description of the browsers can be found in the cited papers.

## 5 Conclusion

In this paper, we present a new formal model for CAESAR systems. In addition, the behavior of the system was also introduced using set theory functions as well as integrated time-space-motion logic. The current Augmented Reality systems can be modeled with mathematical precision by means of the proposed model. The developed

method contributes to the development of theoretical foundations of Augmented Reality systems. For a demonstration of the practical applicability of CAESAR model, two context-aware mobile Augmented Reality applications were briefly presented which are based on our formal model. In the future, a more in-depth investigation into the implementation of logical part of the model is needed.

**Acknowledgments.** This research is supported by Ministry of Culture, Sports and Tourism (MCST) and Korea Creative contents Agency (KOCCA) in the Culture Technology (CT) Research & Development Program 2014.

## References

1. Azuma, R.T.: A survey of augmented reality. *Presence* **6**(4), 355–385 (1997)
2. Schilit, B., Adams, N., Want, R.: Context-aware computing applications. In: *Mobile Computing Systems and Applications*, pp. 85–90 (1994)
3. Milgram, P., Kishino, F.: A taxonomy of mixed reality visual displays. *IEICE Trans. Inf. Syst.* **77**(12), 1321–1329 (1994)
4. Reicher, T.: A framework for dynamically adaptable augmented reality systems. PhD dissertation, Technical University Munich (2004)
5. Galton, A.: Towards an integrated logic of space, time, and motion. *IJCAI* **93**, 1550–1555 (1993)
6. Randell, D.A., Cui, Z., Cohn, A.G.: A spatial logic based on regions and connection. *KR* **92**, 165–176 (1992)
7. Lenzerini, M.: Data integration: a theoretical perspective. In: *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pp. 233–246. ACM (2002)
8. Bernstein, P.A., Rahm, E.: A survey of approaches to automatic schema matching. *VLDB J.* **10**(4), 334–350 (2001)
9. Benjelloun, O., Garcia-Molina, H., Menestrina, D., Su, Q., Whang, S.E., Widom, J.: Swoosh: a generic approach to entity resolution. *VLDB J.* **18**(1), 255–276 (2009)
10. Pérez, J., Arenas, M., Gutierrez, C.: Semantics and complexity of SPARQL. *ACM Trans. Database Syst.* **34**(3), 16 (2009)
11. Volz, R., Kleb, J., Mueller, W.: Towards ontology-based disambiguation of geographical identifiers. In: *I3* (2007)
12. Ricci, F., Rokach, L., Shapira, B., Kantor, P.B.: *Recommender Systems Handbook*. Springer, Heidelberg (2011)
13. Matuszka, T., Kámán, S., Kiss, A.: A semantically enriched augmented reality browser. In: Shumaker, R., Lackey, S. (eds.) *VAMR 2014, Part I. LNCS*, vol. 8525, pp. 375–384. Springer, Heidelberg (2014)
14. Kim, E., Kim, S., Kim, J., Matuszka, T., Park, N., Park, H., Jo, J., Kim, K., Hong, S., Kim, J., Woo, W.: AR reference model in k-culture time machine. In: *Proceedings of HCI International 2016*. Springer International Publishing, Heidelberg (2016)